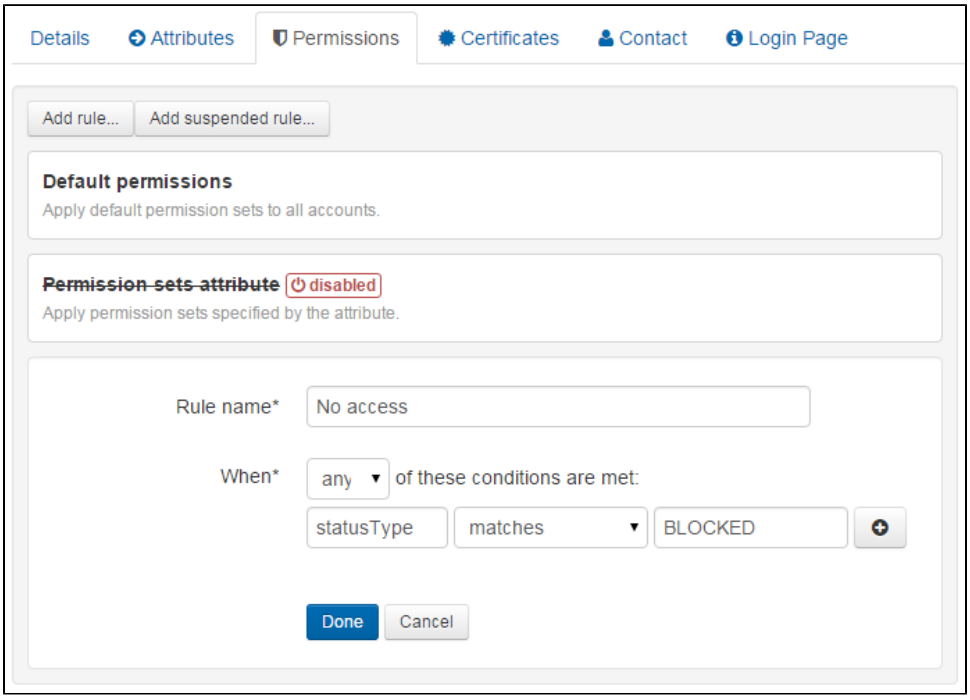
Suspend rules

Under the permissions tab of a local connection you can set up permission set rules and options, and also define rules for suspending a user's access based on attributes passed by the local source - e.g. the `patronStatusInfo.statusType` attribute from a Sirsi system.

You can base the rule on any attribute or combination of attributes supplied by your local system. Just as with the permission set rules, the attributes need only be passed, they do not have to be otherwise mapped.

Accounts that match the suspend rules will not allow access to resources and are shown in the account list as suspended.

1. Click the add suspend rule button
2. Give it a name
3. In the 'When' area:
    a. Enter the name of the attribute in your local source. Attribute names are case sensitive.
    b. Select your condition, e.g. contains, starts with
    c. Enter your comparison
    d. If you need to add other conditions, add additional lines with the plus button and specify if any or all conditions must be met as appropriate. You can add as many lines as you need to.



4. When happy, click done and set up any other rules. You can create as many rules as you need to.
5. Use the save button when done and subsequent logins will use these new rules. (The save button is hidden whilst you are adding or editing rules)

Examples

If you wanted to suspend access for users when *memberOf* contains *Visitor* OR when *memberOf* contains *walk-in* you would specify both conditions and select when *any* conditions are met.

If you wanted to suspend access only when *memberOf* contains *Staff* AND when *memberOf* contains *Teaching* you would specify both conditions and select when *all* conditions are met.

If you wanted to suspend access when users did not have specific attribute values, you would use *all* - e.g. *OU* does not match *Physics_Department* AND *OU* does not match *Chemistry_Department*.

Multi-valued attributes

A multi-valued attribute is one that has several values... at the same time. It's not several values joined together into one value, but several distinct values that are grouped together - the techies would call it an array. An example might be if instead of having different fields for home, work and mobile telephone number, there was one field called telephone numbers that contained any and all phone numbers for that person.

```
<telephoneNumbers>
        <value: "+1 202-456-1414">
        <value: "+44 (0)1 811 8055">
        <value: "+1 636-555-3226">
</telephoneNumbers>
```

The most common multi-valued attribute that is likely to come up is the 'memberOf' attribute from Active directory (or Azure) which contains security and distribution groups.

Where this becomes important is that when a rule looks at a multi-valued attribute, then *all* of the values are treated as discrete when evaluated for match conditions and if *any* of the values meet the condition the rule returns 'true' and does the thing. This is great if you're doing a positive match (e.g. 'contains') because in general if any of the values match you want the rule to apply, however you need to be very careful with negative matches (e.g. 'does not contain') because if any of the values meet that condition then the rule also returns a match - e.g. if your rule is 'does not contain: Staff' and the attribute has three values of ['Full Time Staff', 'Physics Staff', 'Cat appreciation society'], it will return 'true' since one of the values does not contain Staff.

This is one of those computer logic things that isn't obvious at first, but makes sense when you think it through and, once understood, opens many possibilities. In purely practical terms though is is often best to stick with the positive matches such as 'contains' or 'matches' when you're working with multi-valued attributes.